NoSQL Databases

We know that MongoDB is a NoSQL Database, so it is very necessary to know about NoSQL Database to understand MongoDB throughly.

What is NoSQL Database

Databases can be divided in 3 types:

1. RDBMS (Relational Database Management System)
2. OLAP (Online Analytical Processing)
3. NoSQL (recently developed database)

NoSQL Database

NoSQL Database is used to refer a non-SQL or non relational database.

It provides a mechanism for storage and retrieval of data other than tabular relations model used in relational databases. NoSQL database doesn't use tables for storing data. It is generally used to store big data and real-time web applications.

History behind the creation of NoSQL Databases

In the early 1970, Flat File Systems are used. Data were stored in flat files and the biggest problems with flat files are each company implement their own flat files and there are no standards. It is very difficult to store data in the files, retrieve data from files because there is no standard way to store data.

Then the relational database was created by E.F. Codd and these databases answered the question of having no standard way to store data. But later relational database also get a problem that it could not handle big data, due to this problem there was a need of database which can handle every types of problems then NoSQL database was developed.

Advantages of NoSQL

* It supports query language.
* It provides fast performance.
* It provides horizontal scalability.

MongoDB Advantages

* **MongoDB is schema less**. It is a document database in which one collection holds different documents.
* There may be **difference between number of fields, content and size of the document** from one to other.
* **Structure of a single object is clear** in MongoDB.
* There are **no complex joins** in MongoDB.
* MongoDB provides the **facility of deep query** because it supports a powerful dynamic query on documents.
* It is very **easy to scale**.
* It **uses internal memory for storing working sets** and this is the reason of its fast access.

Distinctive features of MongoDB

* Easy to use
* Light Weight
* Extremely faster than RDBMS

Where MongoDB should be used

* Big and complex data
* Mobile and social infrastructure
* Content management and delivery
* User data management
* Data hub

Performance analysis of MongoDB and RDBMS

* In relational database (RDBMS) tables are using as storing elements, while in MongoDB collection is used.
* In the RDBMS, we have multiple schema and in each schema we create tables to store data while, MongoDB is a document oriented database in which data is written in BSON format which is a JSON like format.
* MongoDB is almost 100 times faster than traditional database systems.

|  |  |
| --- | --- |
| **Data Types** | **Description** |
| String | String is the most commonly used datatype. It is used to store data. A string must be UTF 8 valid in mongodb. |
| Integer | Integer is used to store the numeric value. It can be 32 bit or 64 bit depending on the server you are using. |
| Boolean | This datatype is used to store boolean values. It just shows YES/NO values. |
| Double | Double datatype stores floating point values. |
| Min/Max Keys | This datatype compare a value against the lowest and highest bson elements. |
| Arrays | This datatype is used to store a list or multiple values into a single key. |
| Object | Object datatype is used for embedded documents. |
| Null | It is used to store null values. |
| Symbol | It is generally used for languages that use a specific type. |
| Date | This datatype stores the current date or time in unix time format. It makes you possible to specify your own date time by creating object of date and pass the value of date, month, year into it. |

Data Modeling in MongoDB

In MongoDB, data has a flexible schema. It is totally different from SQL database where you had to determine and declare a table's schema before inserting data. MongoDB collections do not enforce document structure.

The main challenge in data modeling is balancing the need of the application, the performance characteristics of the database engine, and the data retrieval patterns.

Consider the following things while designing the schema in MongoDB

* Always design schema according to user requirements.
* Do join on write operations not on read operations.
* Objects which you want to use together, should be combined into one document. Otherwise they should be separated (make sure that there should not be need of joins).
* Optimize your schema for more frequent use cases.
* Do complex aggregation in the schema.
* You should duplicate the data but in a limit, because disc space is cheaper than compute time.

**For example:**

let us take an example of a client who needs a database design for his website. His website has the following requirements:

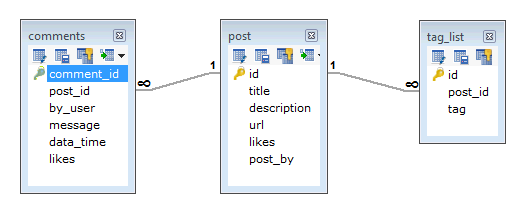
Every post is distinct (contains unique title, description and url).

Every post can have one or more tags.

Every post has the name of its publisher and total number of likes.

Each post can have zero or more comments and the comments must contain user name, message, data-time and likes.

For the above requirement, a minimum of three tables are required in RDBMS.



But in MongoDB, schema design will have one collection post and has the following structure:

{

\_id: POST\_ID

title: TITLE\_OF\_POST,

description: POST\_DESCRIPTION,

by: POST\_BY,

url: URL\_OF\_POST,

tags: [TAG1, TAG2, TAG3],

likes: TOTAL\_LIKES,

comments: [

{

user: 'COMMENT\_BY',

message: TEXT,

datecreated: DATE\_TIME,

like: LIKES

},

{

user: 'COMMENT\_BY',

message: TEST,

dateCreated: DATE\_TIME,

like: LIKES

}}}

Next Topic[MongoDB Create Database](https://www.javatpoint.com/mongodb-create-database)